ERDC/CHL CHETN-IV-104
August 2015

Automated Method to Develop a Clark Synthetic Unit Hydrograph within ArcGIS

by Michael Follum

PURPOSE: This Coastal and Hydraulics Engineering technical note (CHETN) describes an automated method to develop a synthetic unit hydrograph of a small watershed using readily available spatial data products, namely Digital Elevation Model (DEM), and land cover data. A Python script is utilized within ArcGIS (Environmental Systems Research Institute 2011) to quickly and efficiently determine how the spatial features of the watershed affect the runoff of the basin and therefore the unit hydrograph at the outlet of the basin.

BACKGROUND: Rainfall-runoff response within a watershed is a core consideration of hydrologists. The use of unit hydrographs as a way to analyze the rainfall-runoff response was first proposed by Sherman (1932). Since then, many interpretations of the unit hydrograph have been developed that assume runoff is routed through a series of linear reservoirs, including the following: instantaneous unit hydrograph method (Nash 1957), geomorphic instantaneous unit hydrograph method (Rodríguez-Iturbe and Valdés 1979), and the Clark unit hydrograph (CUH) method (Clark 1945). The effectiveness of the unit hydrograph is well evident in hydrologic practice, with the popularity of the CUH method having been increased particularly due to its inclusion in the U.S. Army Corps of Engineers (USACE) Hydrologic Engineering Center (HEC) suite of watershed models, including *HEC-1* (Feldman 1995) and the *Hydrologic Modeling System* (Scharffenberg and Fleming 2006).

Within gauged basins, the unit hydrograph can be developed using observed storm events (precipitation and runoff data required). In basins with little-to-no observed data, a synthetic unit hydrograph can be developed. The CUH is a synthetic unit hydrograph that is developed using a time-area method. To develop a hydrograph using the CUH method, three parameters are required: time of concentration (T_c), storage attenuation coefficient (R), and a time-area histogram (TAH) of the basin. T_c is a measure of the time that it takes for water to flow from the most remote part of the basin to the outlet. Theoretically, from the time rainfall begins, all of the runoff from the entire basin should be contributing to the outlet when the time reaches T_c. R represents the effects of storage on the timing of the outflow from the basin. TAH is a means to represent the area within the basin that drains to the outlet at given time intervals and is the focus of this CHETN.

Previous methods have used GIS to help create a CUH for a basin (Labadie 2014; Maidment et al. 1996; Maidment 1993). To help improve implementation times during emergency scenarios, this document shows how a TAH can be automatically derived using spatially heterogeneous parameters based on topographic (derived from DEM) and land cover characteristics. Because the TAH is calculated based on routing times, T_c is intrinsically included within TAH. For the purposes of this document, any delay in outflow due to R is also assumed to be included within the flow routing scheme. A simple Python script was created that automates the generation of the

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **AUG 2015**	2. REPORT TYPE	3. DATES COVERED **00-00-2015 to 00-00-2015**

4. TITLE AND SUBTITLE **Automated Method to Develop a Clark Synthetic Unit Hydrograph within ArcGIS**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Army Engineer Research and Development Center,,Vicksburg,,MS**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This Coastal and Hydraulics Engineering technical note (CHETN) describes an automated method to develop a synthetic unit hydrograph of a small watershed using readily available spatial data products, namely Digital Elevation Model (DEM), and land cover data. A Python script is utilized within ArcGIS (Environmental Systems Research Institute 2011) to quickly and efficiently determine how the spatial features of the watershed affect the runoff of the basin and therefore the unit hydrograph at the outlet of the basin.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a REPORT **unclassified**	b ABSTRACT **unclassified**	c THIS PAGE **unclassified**	**Same as Report (SAR)**	**13**	

CUH. The Python script (Appendix A) runs within ArcGIS and generates the TAH by calling ArcGIS functions and a simple executable, whose code is shown in Appendix B. The format of this CHETN is to show the methodology on how the Python script works in a step-by-step example. The purpose of this CHETN is to show how readily available spatial data can be used within GIS to create a simple, yet accurate, spatially discretized CUH.

METHODS/EXAMPLE: The test location selected for this CHETN is the New River basin in coastal North Carolina (Figure 1). The basin is approximately 211 km^2 and ranges in elevation between 0.55 m and 36.80 m. As presented in Table 1, the area is predominately evergreen forest, cultivated crops, and woody wetlands. This test location was selected because it is near Marine Corps Base Camp LeJeune, the site of a USACE test project aimed at deployment of fast-response hydrologic models.

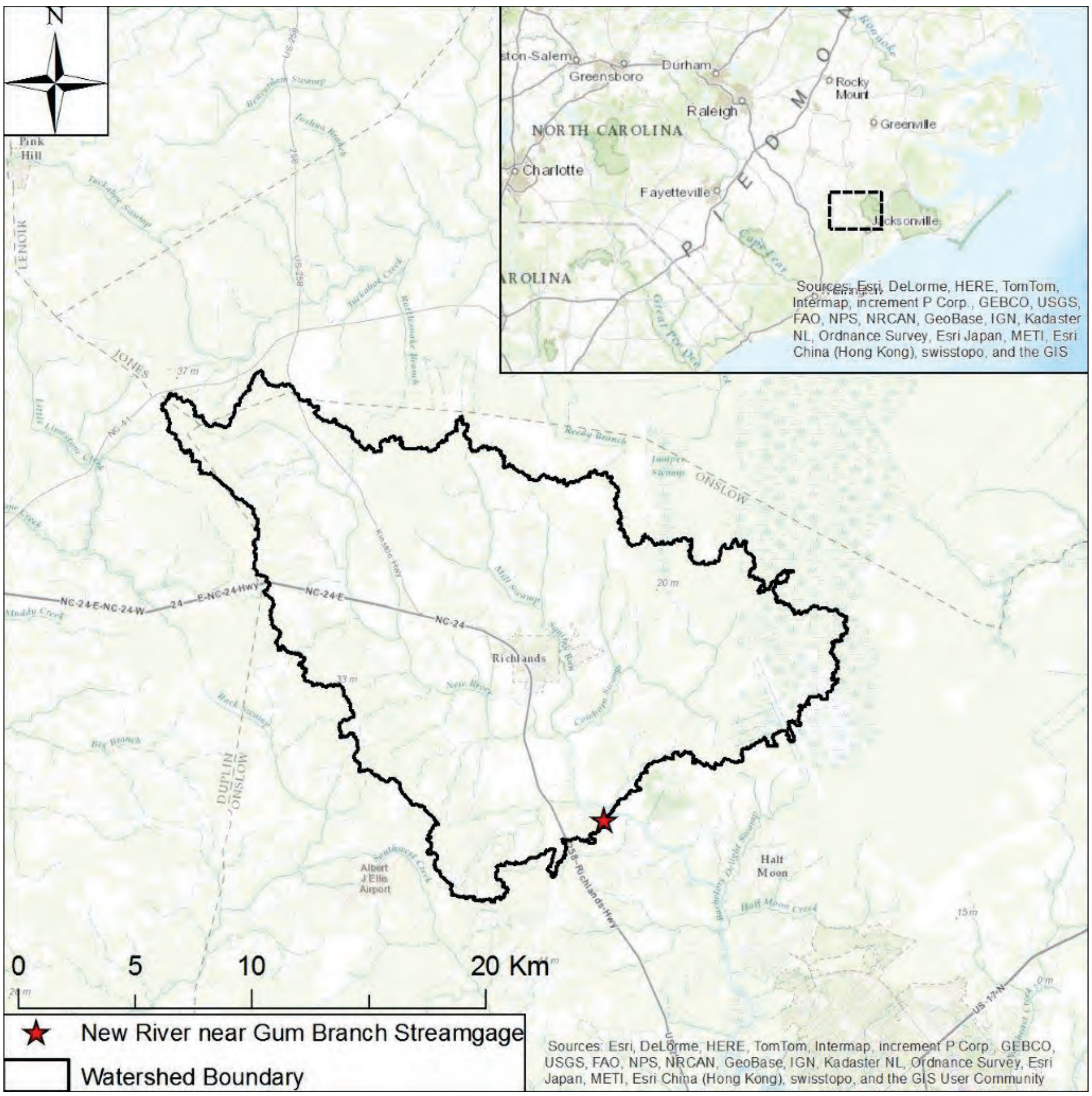

Figure 1. New River basin, NC.

Table 1. NLCD 2006 identification number, classification, associated overland flow conveyance (*K*) values adapted from McCuen (1998), and runoff coefficient (C_r) values taken from the American Society of Civil Engineers (ASCE) and Water Pollution Control Federation (WPCF) (1982).

NLCD 2006 ID	Land Cover	% Area	K	C_r (fraction)
11	Open Water	0.07	25.00	1.00
21	Developed, Open Space	3.84	5.10	0.20
22	Developed, Low Intensity	1.02	10.20	0.30
23	Developed, Medium Intensity	0.25	15.30	0.40
24	Developed, High Intensity	0.01	20.40	0.50
31	Barren Land	0.39	2.00	0.15
41	Deciduous Forest	0.11	1.60	0.15
42	Evergreen Forest	20.84	1.60	0.15
43	Mixed Forest	3.96	1.60	0.15
52	Shrub/Scrub	8.90	7.00	0.15
71	Herbaceuous	2.86	7.00	0.15
81	Hay/Pasture	0.10	3.50	0.15
82	Cultivated Crops	31.82	2.00	0.15
90	Woody Wetlands	25.00	1.60	0.15
95	Emergent Herbaceuous Wetlands	0.81	1.60	0.15

The following is a step-by-step example on how a CUH can be easily derived at a point along the river within the New River basin. The data used within the example are freely and readily available nationwide, so the processes and methods shown can be implemented easily at another location. An example of how flow can be calculated using the CUH method is also shown and compared to observed data.

Step 1. Obtaining spatial data. The 1/3 arc second (~9.645 m) National Elevation Dataset (Gesch et al. 2002) over the project area was obtained from the United States Geological Survey (USGS) (*http://nationalmap.gov/*). A land cover map from the National Land Cover Database 2006 (NLCD 2006) (Fry et al. 2011) was obtained from the same website. Both datasets were reprojected onto a UTM Zone 18N projection (the local UTM projection).

Step 2. Create flow direction and flow accumulation rasters; specify outlet. Flow direction (FDIR) and flow accumulation (FAC) rasters were created over the entire area using the "Hydrology Toolbox" within ArcGIS. For each cell in the domain, FDIR specifies the direction of flow from a cell to its neighboring cell, and FAC indicates how many up-slope cells hydrologically contribute to the cell. Based on the FAC raster, a point shapefile was created at the outlet of the basin being examined. Figure 2 shows the FAC raster as well as the location of the outlet selected. Steps 1 and 2 are typical processes carried out prior to any hydrologic investigation and therefore were not included in the Python script. The remaining steps are carried out within the Python script shown in Appendix A.

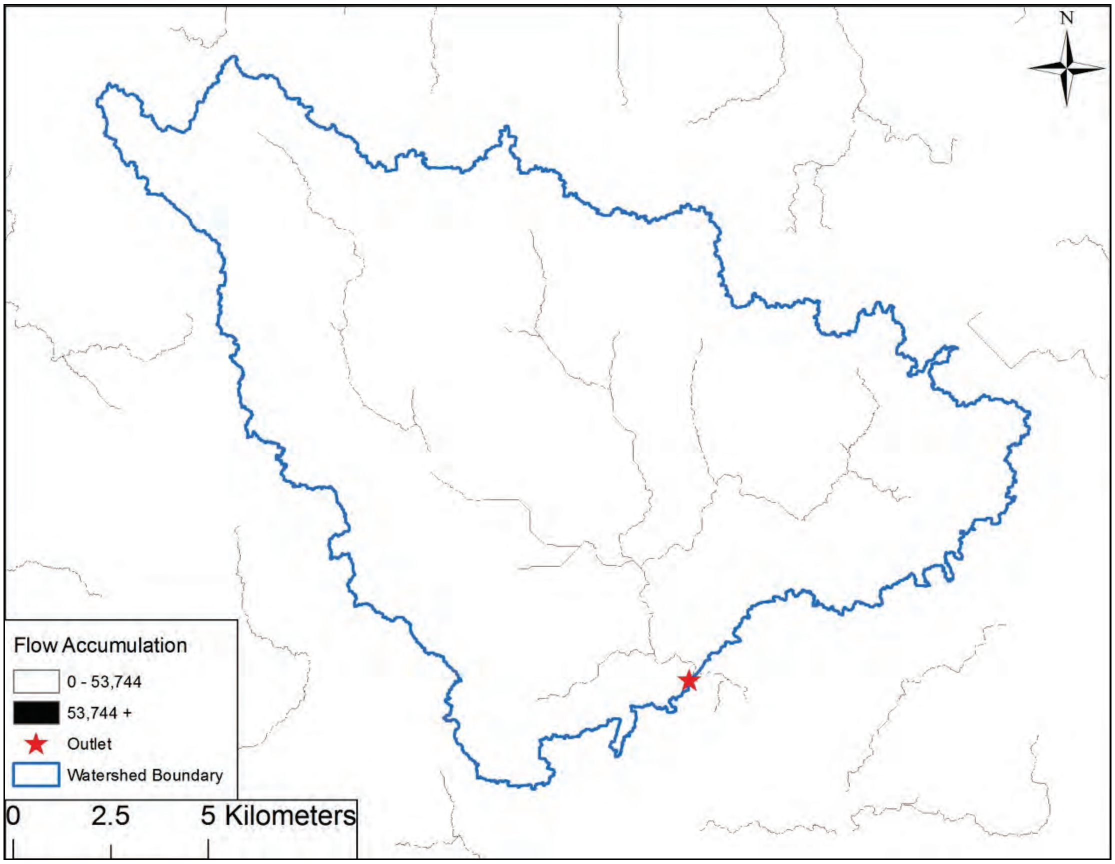

Figure 2. New River FAC, outlet location, and watershed boundary.

Step 3. Creation of TAH. There are two products within this step: (1) a raster that represents the time it takes each cell within the basin to drain to the outlet and (2) a TAH derived from the raster. To complete this, several subtasks must be completed. These tasks are automated within a Python script shown in Appendix A and a C++ code shown in Appendix B. Tasks 1–7 in the following list provide a description of what is simulated within the Python script, while Tasks 8 and 9 provide a brief description of how the two products are ultimately created using a C++ code.

Task 1 (Lines31–41): The watershed is delineated, and a watershed boundary is created (Figure 2). The elevation and flow accumulation rasters are clipped to the watershed boundary. A stream grid is created by assuming that any FAC cell that drains an area greater than 5 km^2 (cell value of 53,744 at a 9.645 m resolution) is considered a stream.

Task 2 (Lines 44–48): The slope of each cell is calculated using the "Slope" function within ArcGIS. The slopes are calculated for an area larger than the watershed boundary and then are clipped to the watershed boundary. The clipped slope raster is initially calculated as a percentage with all slopes less than 0.001% being assigned a value of 0.001%. The slope raster is then converted into a decimal format raster (*OutSlope_Dec*). A third raster is also created to represent the slope (decimal format) in only the channel cells (*Slope_Chan*).

Task 3 (Lines 51–58): The land cover raster is clipped to the watershed boundary and is set to match the spatial extent and resolution of the DEM raster. Flow conveyance values (K) are then mapped to a new raster (*kfactor*) based on the land cover classification and the associated *K* value. Table 1 lists the values of *K* used in this project, which are based on McCuen (1998) and Nicklow et al. (2006).

Task 4 (Lines 61–63): The time it takes for water to flow overland within each cell (*timeoverland*) is calculated based on an approximation of the Manning's equation:

$$timeoverland = \left[\frac{1\,min}{60\,sec}\right]\left[\frac{1\,ft}{0.3048\,m}\right]K^{-1}\,S^{-0.5} = 0.0547\left(C\,K\right)^{-1}S^{-0.5}$$

where *K* is taken from *kfactor*, *S* is the slope of the cell (decimal, taken from *OutSlope_Dec*), and *timeoverland* is in min·m^{-1}. Values of *K* are approximations that incorporate channel/flow geometry and roughness. It is also realistic to assume that *K* varies based on scale. A calibration parameter (*C*) is included to manipulate the value of *K* for a given basin and is set to 0.3 for this example, which effectively increases the time it takes for water to flow over each cell.

Task 5 (Lines 66–69): A raster representing the time of travel within each channel cell (*timechannel*, min m^{-1}) is calculated using the same approach as for overland flow, with the exceptions that *S* is taken from *Slope_Chan*, and a static value of 25.0 is used for *K*. In reality, the *K* value likely varies with each segment of stream, but for this simple example, it is left as a constant for all channel cells. Similar to Task 4, a channel calibration parameter is also employed to manipulate the value of *K* and is also set to 0.3 for this example.

Task 6 (Lines 72–74): A raster called *traveltime* is created that represents the travel times (min·m^{-1}) of both the overland flow and channel flow. If a cell is classified as a channel, the value from *timechannel* is used, otherwise the value is taken from *timeoverland*. After calculation, the *traveltime* raster is converted into an ASCII format (*traveltime.asc*) for use in Tasks 7–9.

Task 7 (Lines 77–78): Although the time-per-distance for water to travel over each cell has been determined in *Task 6*, the path that water takes typically depends on topography and can be approximated as the path it would follow along increasing cell values of the FAC. The accumulated time for water to travel a hydraulically driven path from each cell to the outlet is required to create a TAH. A function within ArcGIS that can complete this task does not exist. Therefore, a program was written in C++ to complete this task and is shown in Appendix B.

Task 8 (Appendix B): A simple executable is called from the Python script. The executable was developed to determine the time it takes for each cell to flow to the outlet based on the flow pathway exhibited by the FAC raster and the time to flow across a given cell in *Task 6*. The program routes the path water takes from each cell within the watershed to the outlet by following a path of increasing FAC values. The time (min) it takes to travel across a single cell is simply the distance across the cell (m) multiplied by the value within the *traveltime.asc* raster (min·m^{-1}). The total time (min) it takes each cell to route to the outlet is the summation of the times along the pathway. This summation of the time is recorded in the *totaltime.asc* raster and is shown for the New River basin in Figure 3. To clarify, the *totaltime.asc* raster specifies the time, in minutes, it takes for water to travel from that cell to the outlet of the basin.

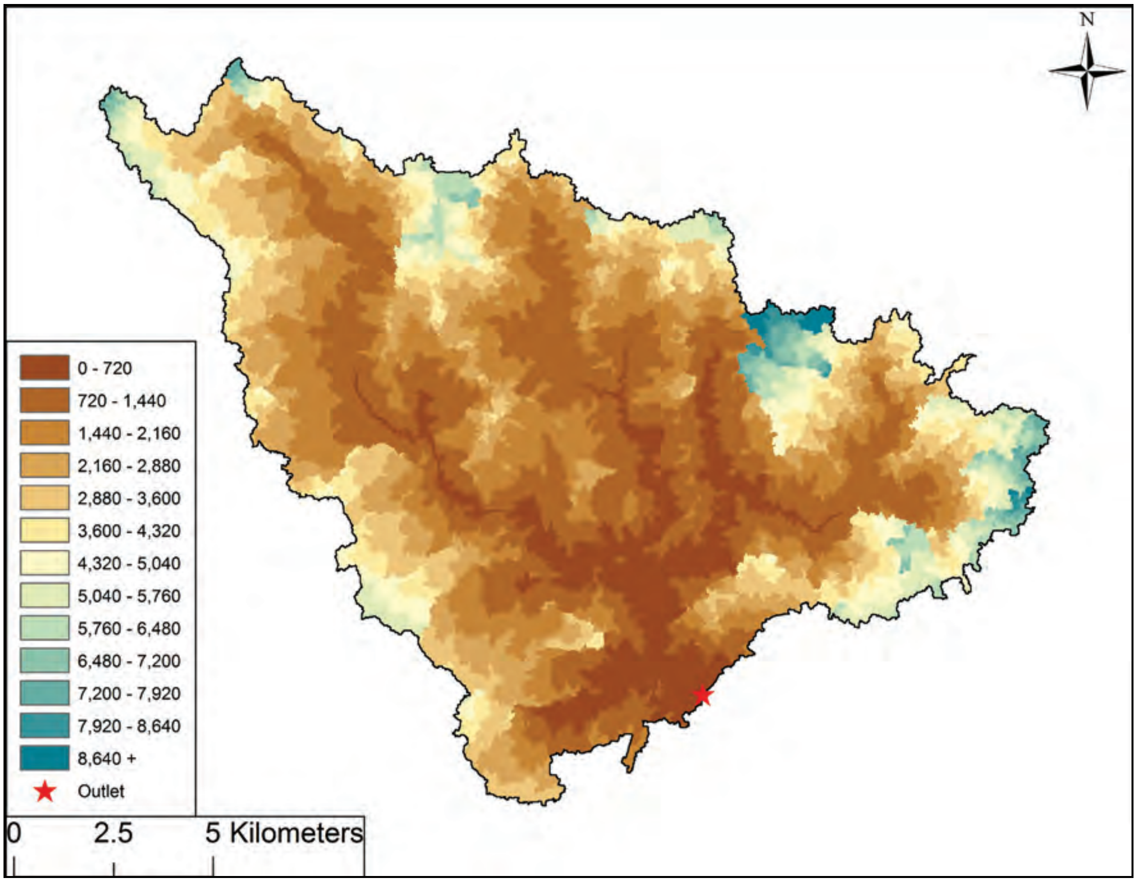

Figure 3. Time in minutes for each cell to drain to outlet (*totaltime.asc*).

Task 9 (Appendix B): A histogram based on the *totaltime.asc* raster shown in *Task 8* can be generated using a simple plotting program, such as Excel. Figure 4 shows such a histogram using 6 hr intervals, which is the 6 hr TAH.

Step 4. Flow Simulation. Two methods are available to determine flow based on a given precipitation. For uniform precipitation, a unit hydrograph is easily generated using the S-curve method and the TAH, where a standard unit hydrograph can be derived for a given time interval of rainfall and used with the rainfall data, or a normalized unit hydrograph can be derived for use in conjunction with peak flow calculations (Bras 1990; Brutsaert 2005). Figure 5 shows an S-curve with a normalized unit hydrograph. The second method, which is used in this document, uses spatial precipitation data in conjunction with the *totaltime.asc* raster to determine how long the rain in each cell takes to get to the outlet. This second method takes a Lagrangian approach to simulate flow because it follows rainfall through each cell to the outlet. Both methods described assume the principle of superposition to simulate outflow.

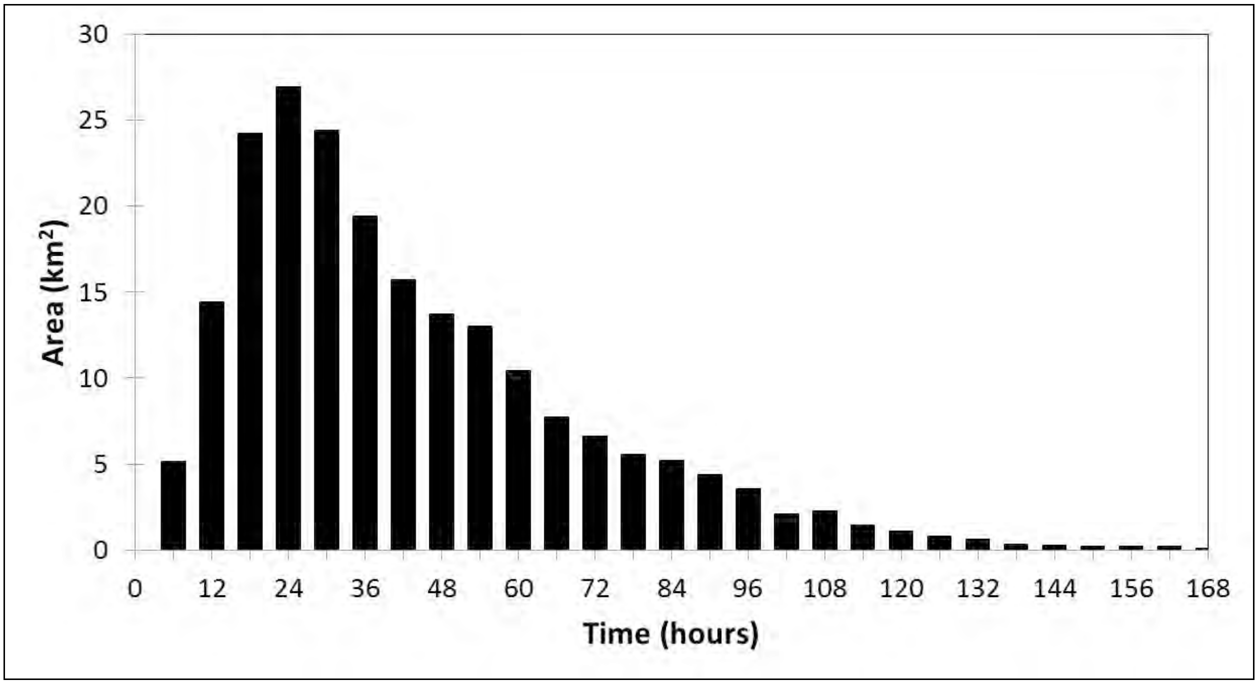

Figure 4. Time-area histogram (TAH) for the New River basin.

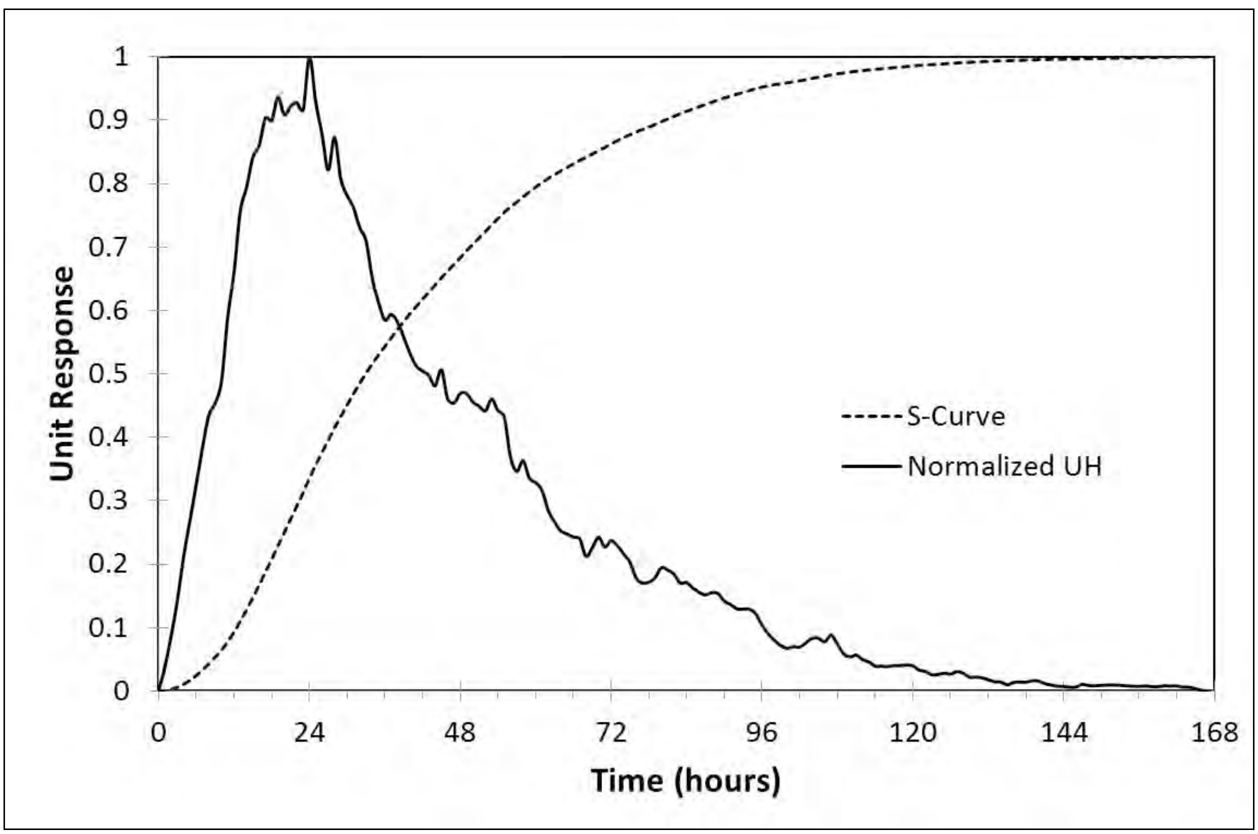

Figure 5. 60 min Clark unit hydrograph (CUH) for the New River basin.

TESTING: Two peak flows from the New River near Gum Branch streamgage occurred between 15 August and 15 September 2011 when multiple large rain events occurred over the study area. This time period will serve to show how the CUH method described can be used to simulate flow.

Next-Generation Radar (NEXRAD) precipitation data in shapefile format were obtained from the National Weather Service (*http://www.srh.noaa.gov/ridge2/RFC_Precip/*). The data were reprojected, clipped to the area of interest, and converted into raster format using the inverse distance weighted (IDW) function within ArcGIS. Flow data from the New River near Gum Branch streamgage were obtained from the USGS (*http://waterdata.usgs.gov/nwis/rt*).

Using a simple programming script (not shown), the effective precipitation (P_{eff}) was applied to the *totaltime.asc* raster to generate a hydraulic response from each hour of precipitation. By combining these hourly responses through the assumption of superposition, a simulated outflow hydrograph is created. P_{eff} represents the fraction of precipitation that contributes to immediate runoff, and is calculated as

$$P_{eff} = P \cdot C_r$$

where P is the precipitation over an hour time period for a given cell from the NEXRAD data and C_r is the runoff coefficient (Table 1). Values of C_r vary widely and are typically calibrated parameters. The values of C_r used in this project are based on land cover classifications and ASCE and WPCF (1982). Developed areas are based on residential and neighborhood values, while all other areas use a value of 0.15, which is the approximate average of the values for lawns, unimproved areas, and parks.

The time-series of hourly precipitation (volume, m^3) applied to the watershed is shown with the simulated (Q_{sim}) and observed flow (Q_{obs}) in Figure 6. Q_{sim} for the first event (26 August–4 September) peaked approximately 7 hr before Q_{obs} and overestimated the peak by approximately 5.51 m^3 s^{-1}. Q_{sim} peaked approximately 9 hr after Q_{obs} during the second event (7–10 September) and underestimated the peak by approximately 4.41 m^3 s^{-1}. The duration of both events matched reasonably well when compared to Q_{obs}. This highlights potential areas of improvement in both the inclusion of baseflow estimation and the methods used to calculate the time to flow across each cell. For both events, Table 2 shows the simulated and observed peak flows, mean bias error (MBE), root mean squared error (RMSE), and Pearson's correlation coefficient (R).

$$MBE = \left[n^{-1} \sum_{i=1}^{n} \left(Q_{sim} - Q_{obs} \right) \right] \qquad RMSE = \left[n^{-1} \sum_{i=1}^{n} \left(Q_{sim} - Q_{obs} \right)^2 \right]^{0.5}$$

Table 2. Flow event statistics (all in units of m^3 s^{-1} except R, which is 0-1).

	Peak Q_{obs}	Peak Q_{sim}	MBE	RMSE	R
Flow Event 1	30.30	35.81	-1.73	3.77	0.95
Flow Event 2	10.34	5.93	-1.14	2.19	0.84

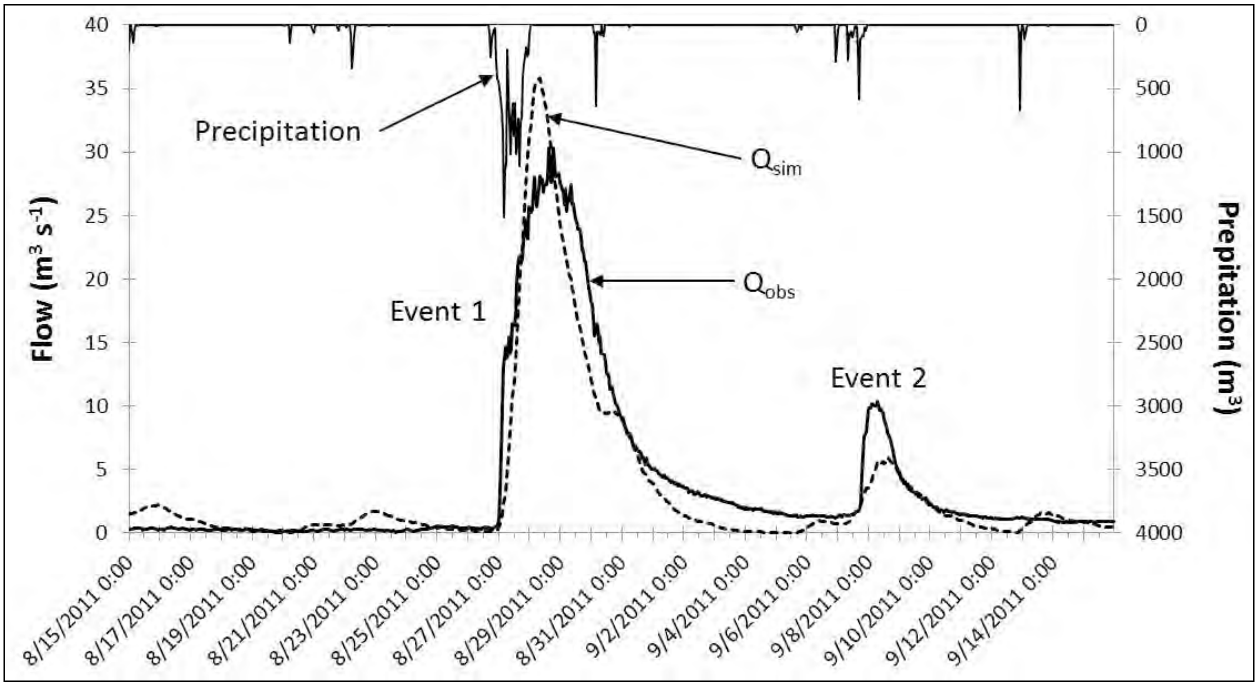

Figure 6. Test events: Precipitation (solid black line, top), Observed Flow (solid black line, bottom), and Simulated Flow (dotted black line).

CONCLUSIONS: This CHETN describes in a step-by-step procedure how to automatically develop a time-area unit hydrograph based only on elevation and land cover data. The CHETN also provides the programming code, which can be used by anyone with low-to-intermediate ArcGIS and programming skills. It was also shown that the time-area unit hydrograph methods can in turn be used to accurately simulate outflow from a small basin when precipitation data is known. Through an example, distributed rainfall data was applied to the generated synthetic unit hydrograph to produce accurate results. Although the methods presented are simple, further refinement can be introduced into the framework to improve accuracy in time-to-drain estimates and effective precipitation estimation. Although this project was developed to assist in a specific military need, the methods may be applicable to other modeling frameworks that use a unit hydrograph method.

POINTS OF CONTACT: This CHETN was prepared by the Military Engineering Program through the Austere Entry Assessment project. The POC for technical inquiries is Mike Follum (*Michael.L.Follum@usace.army.mil*). This technical note should be referenced as follows:

> M. Follum. 2015. *Automated method to develop a Clark synthetic unit hydrograph within ArcGIS*. ERDC/CHL CHETN-IV-104. Vicksburg, MS: U.S. Army Engineer Research and Development Center. An electronic copy of this CHETN is available from *http://chl.erdc.usace.army.mil/chetn*.

REFERENCES

American Society of Civil Engineers (ASCE) and Water Pollution Control Federation (WPCF) (Joint Committee). 1982. Design and construction of sanitary and storm sewers. *ASCE (American Society of Civil Engineers) Manual on Engineering Practice* No. 37 and *WPCF (Water Pollution Control Federation) Manual of Practice* No. 9., 5th ed.

Bras, R. L. 1990. *Hydrology: An introduction to hydrologic science*. Reading: Addison-Wesley.

Brutsaert, W. 2005. *Hydrology: An introduction*. New York: Cambridge University Press.

Clark, C. 1945. Storage and the unit hydrograph. *Transactions of the American Society of Civil Engineers* 110(1):1419–1446.

Environmental Systems Research Institute. 2011. *ArcGIS Desktop: Release 10*. Redlands, CA: Environmental Systems Research Institute.

Feldman, A. 1995. HEC-1 flood hydrograph package. *Computer Models of Watershed Hydrology* 119:150.

Fry, J. A., G. Xian, S. Jin, J. Dewitz, C. Homer, Y. Limin, C. Barnes, N. Herold, and J. Wickham. 2011. Completion of the 2006 national land cover database for the conterminous United States. *Photogrammetric Engineering and Remote Sensing* 77(9):858–864.

Gesch, D., M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, and D. Tyler., 2002. The national elevation dataset. *Photogrammetric Engineering and Remote Sensing* 68(1):5–32.

Labadie, J. 2014. Lecture Handouts for CIVE 577: *GIS in civil and environmental engineering*, Lecture 8: *Time-area unit hydrographs using GIS*. Ft. Collins, CO: Department of Civil and Environmental Engineering, Colorado State University,

Maidment, D., F. Olivera, A. Calver, A. Eatherall, and W. Fraczek., 1996. Unit hydrograph derived from a spatially distributed velocity field. *Hydrological Processes* 10(6):831–844.

Maidment, D. R. 1993. Developing a spatially distributed unit hydrograph by using GIS. In *Applications of Gepographic Information Systems in Hydrology and Water Resources Management*, ed. K. Kovar and H. Nachtnebel. IAHS 211:181–192.

McCuen, R. H. 1998. *Hydrologic analysis and design.* New Jersey: Prentice-Hall.

Nash, J. 1957. The form of the instantaneous unit hydrograph. In *General Assembly of Toronto, Vol. III Surface Waters, Prevision, Evaporation.* IAHS 45(3):114–121.

Nicklow, J. W., P. F. Boulos, M. K. Muleta. 2006. *Comprehensive urban hydrologic modeling handbook for engineers and planners*. MWH Soft, Incorporated.

Rodríguez-Iturbe, I., J. B. Valdés. 1979. The geomorphologic structure of hydrologic response. *Water Resources Research* 15(6):1409–1420.

Scharffenberg, W. A., and M. J. Fleming. 2006. *Hydrologic modeling system HEC-HMS: User's manual*. Davis, CA: U.S. Army Corps of Engineers, Hydrologic Engineering Center.

Sherman, L. K. 1932. Streamflow from rainfall by the unit-graph method. *Engineering News Record* 108:501–505.

Appendix A: Python Program Code Executed through ArcGIS

```
# Import arcpy module
import arcpy
from arcpy import env
from arcpy.sa import *
import arcpy.cartography as CA
import arcpy.management as DM
import arcpy.analysis as ANL
arcpy.CheckOutExtension("spatial")
arcpy.env.workspace = "CUH_Workspace.gdb"                 #Default Workspace (have to create this)
arcpy.env.overwriteOutput = True                          # Overwrite pre-existing files

MF = "C:\\Work_Folder"                                    #This is simply a folder for the output data
Outlet = "C:/Work_Folder/NR_GumBranch_Gage.shp"           #outlet location of the basin
Big_DEM = "C:/Work_Folder/CL_DEM_10m_cl.img"              #DEM of the Area (larger than the actual basin size)
Big_FAC = "C:/Work_Folder/CL_FAC.img"                     #FAC of the Area (larger than the actual basin size)
Big_FDIR = "C:/Work_Folder/CL_FDIR.img"                   #Flow Dir of the Area (larger than the actual basin size)
NLCD = "C:/Work_Folder/NC_NLCD_2006_clipped.tif"          #Land Cover of the Area (larger than the actual basin size)

# Local variables:
Watershed_Poly = MF +"\\" + "Watershed_Poly.shp"
FAC = MF + "\\" + "FAC"
FAC_ASC = MF + "\\" + "FAC_ASC.asc"
kfactor = MF + "\\" + "kfactor"
timeoverland = MF + "\\" + "timeoverland"
timechannel = MF + "\\" + "timechannel"
traveltime = MF + "\\" + "traveltime"
traveltimeasc = MF + "\\" + "traveltime.asc"
LC_Maskasc = MF + "\\" + "LC_Mask.asc"

# Delineate a Watershed and Create one with a mask watershed (Watershed_1)
outSnapPour = SnapPourPoint(Outlet, Big_FAC, 20, "FID")
outWatershed = Watershed(Big_FDIR, outSnapPour)
Watershed_1 = Con(outWatershed >= 0, 1)
arcpy.RasterToPolygon_conversion(Watershed_1, Watershed_Poly, "NO_SIMPLIFY", "VALUE")

Extract_DEM = ExtractByMask(Big_DEM, Watershed_Poly)
Extract_FAC = ExtractByMask(Big_FAC, Watershed_Poly)
Extract_FAC.save(FAC)
arcpy.RasterToASCII_conversion(Extract_FAC, FAC_ASC)
StrmVal = 53744.0                #53744 is the number of cells within 5 km2
ChanGrid = Con(Extract_FAC > StrmVal, 1)

# Calculate Slope of the Basin
OutSlope = Slope(Big_DEM, "PERCENT_RISE", 1)
OutSlope_Clip = ExtractByMask( OutSlope, Watershed_Poly )
OutSlope_Percent = Con(OutSlope_Clip <= 0.001, 0.001, OutSlope)
OutSlope_Dec = Divide(OutSlope_Percent, 100.0)
Slope_Chan = Times(OutSlope_Dec, ChanGrid)

# Land Use Grid
tempEnvironment0 = arcpy.env.cellSize
arcpy.env.cellSize = "MINOF"
LU_0 = ExtractByMask(NLCD, Watershed_1)
arcpy.env.cellSize = tempEnvironment0
LU_1 = Con(LU_0 == 11,25.0, Con(LU_0 == 21,5.1, Con(LU_0 == 22,10.2, Con(LU_0 == 23,15.3, Con(LU_0 == 24,20.4,
         Con(LU_0 == 41,1.6, Con(LU_0 == 42,1.6, Con(LU_0 == 43,1.6, Con(LU_0 == 52,7.0, Con(LU_0 == 71,7.0,
         Con(LU_0 == 81,3.5, Con(LU_0 == 82,2.0, Con(LU_0 == 90,1.6, Con(LU_0 == 95,1.6, 2.0))))))))))))))
LU_1.save(kfactor)

# Calculate Time (min) and Velocity (m/s) of Overland Flow
OF_Calib = 0.3            #Overland Flow Calibration Factor
OV_2 = 0.0547 * (1.0/(OF_Calib*LU_1)) * Power(OutSlope_Dec, -0.5)
OV_2.save(timeoverland)

# Get the Slope of the Channel and a Time of Travel within the Channel (minutes)
K_Channel = 25.0          #Conveyance of channel, assumed to be 25
CH_Calib = 0.3            #Channel Flow Calibration Factor
Chan_1 = 0.0547 * (1/(K_Channel*CH_Calib)) * Power(Slope_Chan, -0.5)
Chan_1.save(timechannel)

# Compute Travel Time
Time_0 = Con(IsNull(Chan_1),OV_2,Chan_1)
Time_0.save(traveltime)
arcpy.RasterToASCII_conversion(Time_0, traveltimeasc)

# Run the Calculator of Total Time
import subprocess
subprocess.call('C:\\Work_Folder\\Clark_UH_TotalTime_FAC.exe')
```

Appendix B: C++ Code to Compute TAH

```
#include <stdio.h>
#include <string.h>
#include <math.h>
void main(void){
    int r, c, ro, co, ro1, co1, ro2, co2, ncols, nrows, max_fac;
    double h, distance, val, xll, yll, cellsize, ND, t=60.0;
    char JUNK[1024];
    bool quit=false;

    //File Names
    char LN_Name[1024], OF_Name[1024], TT_Name[1024], Folder_Name[]="C:\\Work_Folder\\";
    strcpy(LN_Name, Folder_Name); strcat(LN_Name,"fac_asc.asc");
    strcpy(OF_Name, Folder_Name); strcat(OF_Name,"totaltime.asc");
    strcpy(TT_Name, Folder_Name); strcat(TT_Name,"traveltime.asc");
    FILE *LN, *OF, *TT;

    //Open Input and Output Files
    if( (LN=fopen(LN_Name, "r")) != NULL ){ printf("\n\nOpened %s", LN_Name); }
    else{ printf("\n\nCould Not Open %s,\n   Please Start OVER!!!", LN_Name ); scanf("%s", &JUNK); return; }
    if( (TT=fopen(TT_Name, "r")) != NULL ){ printf("\nOpened %s", TT_Name); }
    else{ printf("\n\nCould Not Open %s,\n   Please Start OVER!!!", TT_Name ); scanf("%s", &JUNK); return; }
    OF=fopen(OF_Name, "w");

    //Read and Write Header
    fscanf(LN, "%s" "%d", &JUNK, &ncols); fscanf(TT, "%s" "%d", &JUNK, &ncols); fprintf(OF, "%s %d\n", JUNK, ncols);
    fscanf(LN, "%s" "%d", &JUNK, &nrows); fscanf(TT, "%s" "%d", &JUNK, &nrows); fprintf(OF, "%s %d\n", JUNK, nrows);
    fscanf(LN, "%s" "%lf", &JUNK, &xll); fscanf(TT, "%s" "%lf", &JUNK, &xll); fprintf(OF, "%s   %lf\n", JUNK, xll);
    fscanf(LN, "%s" "%lf", &JUNK, &yll); fscanf(TT, "%s" "%lf", &JUNK, &yll); fprintf(OF, "%s   %lf\n", JUNK, yll);
    fscanf(LN, "%s" "%lf", &JUNK, &cellsize); fscanf(TT, "%s" "%lf", &JUNK, &cellsize); fprintf(OF, "%s %lf\n", JUNK, cellsize);
    fscanf(LN, "%s" "%lf", &JUNK, &ND); fscanf(TT, "%s" "%lf", &JUNK, &ND); fprintf(OF, "%s %lf", JUNK, ND);

    //Allocate memory and read in the data
    printf("\nReading through the Raster Files\n");
    float **T = new float*[nrows];       //Total Time (min)
    double **R = new double*[nrows];     //Rate of Time (min/m)
    int **D = new int*[nrows];           //Flow Accumulation Value
    for(ro=0; ro<nrows; ro++){
        T[ro] = new float[ncols];  R[ro] = new double[ncols];  D[ro] = new int[ncols];
        for(co=0; co<ncols; co++){
            T[ro][co]=-1.0;
            fscanf(TT,"%lf",&R[ro][co]); fscanf(LN,"%lf",&val);  D[ro][co] = int(val+0.01); } }

    //Go through the Each Cell
    printf("\nLooking at each cell and analyzing the Total Time\n");
    for(ro=0; ro<nrows; ro++){
        fprintf(OF,"\n");
        for(co=0; co<ncols; co++){
            if(D[ro][co]<0){ T[ro][co]=0.0;  fprintf(OF,"%f ",-9999.0); continue; }
            quit=false; val=0.0; distance=0.0; ro1=ro; co1=co;
            while(quit==false){
                //Look Diagonals First
                max_fac = D[ro1][co1];
                r=(ro1); c=(co1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=(ro1-1); c=(co1-1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=(ro1-1); c=(co1+1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=(ro1+1); c=(co1-1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=(ro1+1); c=(co1+1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}

                //Now Look at Cardinal Directions (preferred if there is a tie between this and a diagonal direction
                r=(ro1-1); c=co1; if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=ro1; c=(co1-1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=ro1; c=(co1+1); if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}
                r=(ro1+1); c=co1; if(r>=0 && c>=0 && c<ncols && r<nrows){if( D[r][c]>=max_fac){max_fac=D[r][c]; ro2=r; co2=c;}}

                //Determine the distance across the cell
                if( ro1==ro2 || co1==co2 ){ h=1.0; }    //This means the adjacent cell is in a cardinal direction
                else{ h=sqrt(2.0); }    //This means the adjacent cell is at a diagonal
                h = h*cellsize;     distance = distance + h;

                if( co1>=ncols || ro1>=nrows || co1<0 || ro1<0 ){quit=true;}
                //This is the source cell, so this is the end of the line
                else if(D[ro2][co2]==D[ro1][co1]){val = val + R[ro1][co1]*h; quit=true; }
                //This cell has already been analyzed, so just use it's value
                else if( T[ro2][co2] > 0.0 ){ val = val + 0.5*h*(R[ro1][co1]+R[ro2][co2]) + T[ro2][co2]; quit = true; }
                else{ val = val + 0.5*h*(R[ro1][co1]+R[ro2][co2]); co1=co2; ro1=ro2; }  }
            T[ro][co] = val; fprintf(OF,"%f ",T[ro][co]);  }  }
    fclose(LN); fclose(OF); fclose(TT);
    return; }
```

NOTE: The contents *of this technical note are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such products.*